LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-654916

Code Coupling via Jacobian-Free Newton-Krylov Algorithms with Application to Magnetized Fluid Plasma and Kinetic Neutral Models

I. Joseph

May 27, 2014

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Code Coupling via Jacobian-Free Newton-Krylov Algorithms with Application to Magnetized Fluid Plasma and Kinetic Neutral Models LLNL-TR-654916

Ilon Joseph*
Lawrence Livermore National Laboratory
(Dated: May 23, 2014)

Jacobian-free Newton-Krylov (JFNK) algorithms are a potentially powerful class of methods for solving the problem of coupling codes that address different physics models. As communication capability between individual submodules varies, different choices of coupling algorithms are required. The more communication that is available, the more possible it becomes to exploit the simple sparsity pattern of the Jacobian, albeit of a large system. The less communication that is available, the more dense the Jacobian matrices become and new types of preconditioners must be sought to efficiently take large time steps. In general, methods that use constrained or reduced subsystems can offer a compromise in complexity. The specific problem of coupling a fluid plasma code to a kinetic neutrals code is discussed as an example.

*E-mail address: joseph5@llnl.gov

I. THE CODE COUPLING PROBLEM

Complex multiscale physics problems are being attacked at the forefront of computational science. Often, there are highly efficient specialized algorithms for treating a piece of the problem rather efficiently which have been implemented into sophisticated numerical tools. However, when treating the coupled problem, the efficiency is lost due to fast timescales in the coupling terms and their interaction with fast internal timescales. When long time scale solutions are sought, e.g. near equilibrium, the coupling becomes prohibitive. Moreover, the problems of interest are typically nonlinear and may possess multiple solution branches. If an accurate solution is required on fast time scales, then these fast time scales must be resolved. If only the longer time scale solutions are sought, then an effective means to treat this coupling problem is to use a nonlinear solver. The more computationally expensive each call to the individual codes are and the more time-consuming it is to exchange information between codes, the more important it becomes to use a highly efficient nonlinear solver that is scalable to large problem sizes.

Jacobian-free Newton-Krylov solvers [1, 2] are a powerful class of efficient scalable solvers that may be optimal for these scenarios. In the next section, we consider the way in which communication between the nonlinear solver and the individual codes affects the solution algorithm. The communication level varies between "transparent" where the entire system is treated as a whole and the subalgorithms are only used to evaluate equations, to "opaque" where the subalgorithms can only be used to provide the solution for fixed right hand side. For practical purposes, a compromise is likely to be required. When one subalgorithm dominates the other in complexity, it may be necessary to use the "semi-transparent" subalgorithms defined below which attempt to constrain part of the solution. This has the disadvantage of introducing more complex couplings between subalgorithms.

In Section III, we consider the specific problem of coupling a fluid plasma code [4–6] to a kinetic neutrals code [3]. Today, Monte-Carlo (MC) solvers for the kinetic neutrals problem [7, 8] are more widely used. One of the important issues introduced by the MC approach is statistical noise. If the equation evaluations and Jacobian calculations are subject to noise, they will become ineffective at generating useful results unless statistical uncertainty is sufficiently reduced. Determination of the optimal trade-off between time-step and noise is a complex research subject of its own and is beyond the scope of the following considerations. The final sections summarizes the conclusions.

II. CODE COUPLING ALGORITHMS

A. JFNK

If it is possible to invest in an overarching nonlinear solver, then Jacobian-free Newton-Krylov solvers [1, 2] are a powerful class of efficient scalable solvers. Let the n unknowns be $\mathbf{u} = \{u^k\}$ and the nonlinear equations to be solved $\mathcal{N}^k(\mathbf{u}) = 0$ define the global system. The Newton root finding strategy solves the equations $\mathcal{N}(\mathbf{u}) = 0$ via iteration for a sequence $\{\mathbf{u}_{(j)}\}$ that converges to the solution. The Newton iteration is

$$-\mathbf{J}_{(j)} \cdot \Delta\mathbf{u}_{(j+1)} = \mathcal{N}_{(j)} \equiv \mathcal{N}(\mathbf{u}_{(j)}). \tag{1}$$

It is clear that if the solution converges to $LHS = 0$, then the equations have been solved $RHS = 0$. The Jacobian is evaluated, e.g. from the Taylor series, via

$$(\mathbf{J}_{(j)})^k_i \equiv \partial_{\mathbf{u}^i}\mathcal{N}^k_{(j)}\Big|_{u_{(j)}} \simeq \left(\mathcal{N}(\mathbf{u} + \epsilon\hat{\mathbf{e}}^i) - \mathcal{N}(\mathbf{u})\right)/\epsilon \tag{2}$$

where $\hat{\mathbf{e}}^i$ is the i^{th} direction vector and the RHS is a numerical approximation which holds in the limit $\epsilon \to 0$.

The Krylov algorithm allows one to never explicitly form the Jacobian, but only to iteratively evaluate products of the Jacobian and a trial solution. Given initial guess $\mathbf{u}_0$, the vector $\mathbf{J} \cdot \mathbf{u}_0 = 0$, the solution is sought in the subspace $\mathcal{M}^m = \mathrm{span}\{\oplus^{\mathrm{m}}_{\mathrm{j}=0}\mathbf{J}^{\mathrm{j}}\mathbf{u}_0\}$. If the number of Krylov iterations $m = n$, then the method is equivalent to direct inversion. In order for iteration to converge in faster than n iterations, it is necessary to find a good preconditioner for the Jacobian. Thus, the key to this method is the determination of an appropriate preconditioner for the problem at hand.

B. Transparent Coupling: Full JFNK

The "transparent coupling problem" is defined by transparency of communication between subalgorithms: (i) all $\mathbf{u}$ data are accessible and (ii) all equation results $\mathcal{N}(\mathbf{u})$ can be evaluated and passed to the solver. The codes are

only called to initialize the problem and to provide right hand sides. This is likely to be the most computationally efficient and flexible solution strategy because the solver has direct control over all variables. Any reductions that are performed can be designed to act as an efficient preconditioning strategy. The otherwise arbitrary splitting of variables between subalgorithms can complicate the couplings and interaction terms in unforseen ways.

This approach is equivalent to treating the subsystems as part of a whole. However, this approach may be quite demanding in terms of the human time needed to address the problem in this manner. Also, the division into subalgorithms is often based on a good understanding of the processes to be modeled, and if properly interpreted, may already be quite close to a desirable preconditioning for the coupled problem for a large fraction of the domain.

C. Semi-Transparent Coupling: Constrained JFNK

Assume now that data and equation results from one of the codes $\mathfrak{A}$ is accessible (system $\mathcal{A}$, subscript A), but that is too difficult to access necessary information from the other code $\mathfrak{D}$ (system $\mathcal{D}$, subscript D). The total system is divided into n_A variables $\mathbf{u}_A$ and n_D variables $\mathbf{u}_D$. If couplings are neglected, the nonlinear equations for each subalgorithm are $\mathcal{N}_A(\mathbf{u}_A) = 0$ and $\mathcal{N}_D(\mathbf{u}_D) = 0$. When couplings are included, the equations become

$$\mathcal{N}_A(\mathbf{u}_A) + \mathcal{N}_B(\mathbf{u}_A, \mathbf{u}_D) = 0 \tag{3}$$
$$\mathcal{N}_D(\mathbf{u}_D) + \mathcal{N}_C(\mathbf{u}_A, \mathbf{u}_D) = 0. \tag{4}$$

The Jacobian is then split into four subblocks

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \tag{5}$$

where

$$\mathbf{A} = \partial_A(\mathcal{N}_A + \mathcal{N}_B) \tag{6}$$
$$\mathbf{B} = \partial_D \mathcal{N}_B \tag{7}$$
$$\mathbf{C} = \partial_A \mathcal{N}_C \tag{8}$$
$$\mathbf{D} = \partial_D(\mathcal{N}_C + \mathcal{N}_D). \tag{9}$$

Assume that subalgorithm $\mathfrak{D}$ can provide the solution to Eq. 4 for any fixed $\mathbf{u}_A$. Then, these equations can be interpreted as constraints $\mathbf{u}_D(\mathbf{u}_A)$ and JFNK can be implemented on

$$\mathcal{N}_A(\mathbf{u}_A) + \mathcal{N}_B(\mathbf{u}_A, \mathbf{u}_D(\mathbf{u}_A)) = 0. \tag{10}$$

It is clear that (i) computing these equations is at least as expensive as solving for $\mathbf{u}_D(\mathbf{u}_A)$ and (ii) the Jacobian for Eq. 3 alone will be more dense due to the new interaction terms:

$$\mathbf{J}|_{\mathcal{D}} = \hat{\mathbf{A}} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}. \tag{11}$$

Thus, the preconditioner for the constrained system $\mathcal{A}/\mathcal{D}$ must now address the dynamics of both subsystems.

D. Semi-Transparent Coupling: Reduced JFNK

Now assume that the coupling terms $\mathcal{N}_B$ only depend on a small subset of system $\mathcal{D}$, called $\mathcal{E}$ with variables $\mathbf{u}_E$, so that $\mathcal{N}_B(\mathbf{u}_A, \mathbf{u}_E)$. In general, the coupling between $\mathcal{E}$ and $\mathcal{D}/\mathcal{E}$ is strong so that it is not possible to solve only $\mathcal{A}+\mathcal{E}$ system. However, it may be possible to modify subalgorithm $\mathfrak{D}$ to a new version $\mathfrak{F}$ in order to give the JFNK solver explicit control over the state of $\mathcal{E}$. In other words, we assume that the equations $\mathcal{N}_D + \mathcal{N}_C$ can be subdivided into equation sets $\mathcal{N}_{D'} + \mathcal{N}_{C'}$ and $\mathcal{N}_F$. The modified algorithm $\mathfrak{F}$ solves the constraints

$$\mathcal{N}_F(\mathbf{u}_F, \mathbf{u}_E, \mathbf{u}_A) = 0 \tag{12}$$

for any $\mathbf{u}_E, \mathbf{u}_A$. It is clear that the Jacobian $\mathbf{H} = \partial_E \mathcal{N}_E$ must be non-degenerate for a solution to be determined. Then, the system of equations that will be solved via JFNK is

$$\mathcal{N}_A(\mathbf{u}_A) + \mathcal{N}_B(\mathbf{u}_A, \mathbf{u}_E) = 0 \tag{13}$$
$$\mathcal{N}_E(\mathbf{u}_E, \mathbf{u}_F(\mathbf{u}_E)) + \mathcal{N}_{C'}(\mathbf{u}_E, \mathbf{u}_F(\mathbf{u}_E), \mathbf{u}_A) = 0. \tag{14}$$

While the equations are likely to be as expensive to compute as the previous formulation, the advantage of this formulation is that it may be possible to converge in fewer iterations if an efficient preconditioner is found.

The Jacobian of the full system now has the form

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & 0 \\ \mathbf{C}' & \mathbf{D}' & \mathbf{E} \\ \mathbf{F} & \mathbf{G} & \mathbf{H} \end{bmatrix} \tag{15}$$

where $\mathbf{A}, \mathbf{B}$ are still defined by Eq. 6-7 and

$$\mathbf{C}' = \partial_A(\mathcal{N}_{D'} + \mathcal{N}_{C'}) \tag{16}$$
$$\mathbf{D}' = \partial_E(\mathcal{N}_{D'} + \mathcal{N}_{C'}) \tag{17}$$
$$\mathbf{E} = \partial_F(\mathcal{N}_{D'} + \mathcal{N}_{C'}) \tag{18}$$
$$\mathbf{F} = \partial_A \mathcal{N}_F \tag{19}$$
$$\mathbf{G} = \partial_E \mathcal{N}_F \tag{20}$$
$$\mathbf{H} = \partial_F \mathcal{N}_F. \tag{21}$$

The Jacobian of the reduced system has the form

$$\mathbf{J}|_{\mathcal{F}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \hat{\mathbf{C}}' & \hat{\mathbf{D}}' \end{bmatrix} \tag{22}$$

where

$$\hat{\mathbf{C}} = \mathbf{C}' - \mathbf{E}\mathbf{H}^{-1}\mathbf{F} \tag{23}$$
$$\hat{\mathbf{D}} = \mathbf{D}' - \mathbf{E}\mathbf{H}^{-1}\mathbf{G}. \tag{24}$$

Thus, the $\mathbf{A}, \mathbf{B}$ components for $\mathcal{A} - \mathcal{E}$ coupling retain the same sparsity pattern as for the full JFNK solve. The complexity that must be addressed is the more complicated Jacobian in the $\hat{\mathbf{C}}$ and $\hat{\mathbf{D}}$ sectors due to $\mathcal{E} - \mathcal{F}$ coupling. While this may still be challenging to solve, the assumption of the reduced formulation is that $\mathcal{E} - \mathcal{F}$ coupling is a much easier to address than full $\mathcal{A} - \mathcal{D}$ coupling in Eq. 11.

E. Opaque Coupling: Newton Iteration

Now we assume that the internal results can not be accessed from either code, but that solutions to each set of Eqs. 3 (4) can be found for fixed values of $\mathbf{u}_D$ ($\mathbf{u}_A$) respectively. We would still like to perform the Newton iteration $J\delta\mathbf{u} = -\mathcal{N}$, but the lack of communication strictly enforces the block-structure of the Jacobian. The exact inverse of the block matrix in Eq. 5 is

$$\mathbf{J}^{-1} = \begin{bmatrix} \hat{\mathbf{A}}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\hat{\mathbf{D}}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}\hat{\mathbf{A}}^{-1} & \hat{\mathbf{D}}^{-1} \end{bmatrix} \tag{25}$$

where

$$\hat{\mathbf{A}} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C} \tag{26}$$
$$\hat{\mathbf{D}} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}. \tag{27}$$

Thus, if the inverses of the submatrices above can be determined, a full Newton step can still be taken. It is clear that in this case, the inverses of $\hat{\mathbf{A}}, \hat{\mathbf{D}}$ can only be determined by iteration between subalgorithms $\mathfrak{A}$, $\mathfrak{D}$. This solution must be performed 4 times for each iteration: 2 for $\hat{\mathbf{A}}$ and 2 for $\hat{\mathbf{D}}$. In order to reduce the number of iterations, which are presumably costly, it is best if the algorithms can be modified to a form $\mathfrak{A}'$, $\mathfrak{D}'$ that includes a preconditioner for the other equation set so that it is possible to easily evaluate a modified Jacobian $\hat{\mathbf{A}}_0 \simeq \hat{\mathbf{A}}, \hat{\mathbf{D}}_0 \simeq \hat{\mathbf{D}}$. Then the exact inverse

$$\hat{\mathbf{A}}^{-1} \cdot \mathbf{u}_A = \sum_{j=0}^{m} (1 - \hat{\mathbf{A}}_0^{-1}\hat{\mathbf{A}})^j \hat{\mathbf{A}}_0^{-1} \cdot \mathbf{u}_A \tag{28}$$

can be determined by relatively few iterations between codes.

Let us describe this process in more detail. In practice, the Jacobian terms are found by numerical differentiation, e.g. $\mathbf{A}$ is defined via a numerical approximation of $\partial_A \mathcal{N}_A$. Thus, the equation $\mathbf{Ax} = \mathbf{r}$ is actually

$$\mathcal{N}_A(\mathbf{u}_{(j)} + \epsilon \mathbf{x}) - \mathcal{N}_A(\mathbf{u}_{(j)}) = \epsilon \mathbf{r}. \tag{29}$$

If the equations $\mathcal{N}_A(\mathbf{x}) = \mathbf{r}$ are solvable by algorithm $\mathfrak{A}$, then, Eq. 29 can be solved as well. The evaluation of the interaction term $\mathbf{x} = \mathbf{BD}^{-1}\mathbf{Cr}$ at step $\mathbf{u}_{(j)}$ is performed via the following algorithm: (i) using $\mathfrak{D}$ evaluate $\mathbf{x}_1$

$$\mathbf{x}_1 \equiv (\mathcal{N}_C + \mathcal{N}_D)|_{\mathbf{u}_{(j)}}^{\mathbf{u}_{(j)}+\epsilon\mathbf{r}} \tag{30}$$

(ii) and solve for $\mathbf{x}_2$

$$\mathcal{N}_D(\mathbf{u}_{(j)} + \epsilon \mathbf{x}_2) - \mathcal{N}_D(\mathbf{u}_{(j)}) = \mathbf{x}_1. \tag{31}$$

(iii) Then, using $\mathfrak{A}$ evaluate the full step $\mathbf{x}$

$$\mathbf{x} = \mathcal{N}_B(\mathbf{u}_{(j)} + \epsilon \mathbf{x}_2) - \mathcal{N}_B(\mathbf{u}_{(j)}). \tag{32}$$

In order to form a nonlinear preconditioner for iteration, we assume that the equation set can be modified to the form $\mathcal{N} = \mathcal{N}_0 + \mathcal{N}_1$, where (i) $\mathcal{N}_0$ can be solved with minor modifications of the original algorithms $\mathfrak{A}, \mathfrak{D}$ and where $\hat{\mathbf{A}}_0 = \partial_A \mathcal{N}_0$ and $\hat{\mathbf{D}}_0 = \partial_D \mathcal{N}_0$ represent efficient approximations to $\hat{\mathbf{A}}, \hat{\mathbf{D}}$ respectively. In this case, the solution to $\hat{\mathbf{A}}_0 \mathbf{x} = \mathbf{r}$ is performed by using $\mathfrak{A}'$ to solve $\mathcal{E}_0(\mathbf{u}_{(j)} + \epsilon \mathbf{x}) = \epsilon \mathbf{r} + \mathcal{E}_0(\mathbf{u}_{(j)})$.

III. EXAMPLE: COUPLING FLUID PLASMA EQUATIONS TO KINETIC NEUTRAL EQUATIONS

A. Implicit Time Integration

Here we consider the example of coupling a magnetized plasma fluid model [4–6] to a kinetic neutral physics model [3, 7, 8]. Since Monte-Carlo (MC) approaches are often applied to the kinetic neutral problem, the data and equations are often not in a form that is easily accessible to the nonlinear solver, so that the coupling problem is, at best, semi-transparent. The MC code solves the kinetic equation by following a large number of Lagrangian grid points in time. In all cases, this data must be remapped to the grid of interest to the plasma solver.

The equations are assumed to be presentable as a first order in time partial-differential system

$$\mathcal{N} = \partial_t \mathbf{u} + \mathcal{F}(\mathbf{u}) = 0 \tag{33}$$

as well as possible constraint equations $\Delta(\mathbf{u}, \mathbf{u}_\Delta)$ that are relatively easy to solve, so that they need not be considered further. The goal is to solve these equations implicitly in time. For example, the implicit midpoint rule yields

$$\mathcal{N}(\mathbf{u}_{(i+1)}) = \Delta \mathbf{u}_{(i+1/2)} + \Delta t \mathcal{F}(\bar{\mathbf{u}}_{(i+1/2)}) = 0 \tag{34}$$

where

$$\Delta \mathbf{u}_{(i+1/2)} \equiv \mathbf{u}_{(i+1)} - \mathbf{u}_{(i)} \tag{35}$$

$$\bar{\mathbf{u}}_{(i+1/2)} \equiv (\mathbf{u}_{(i+1)} + \mathbf{u}_{(i)})/2 \tag{36}$$

for each time step. It is clear that reducing the time step sufficiently causes the first term to become dominant, at which point the system can be solved by explicit iteration. The goal of taking a large time step requires efficient methods for solving the full nonlinear set of Eq. 34. Each iteration j of the JFNK solver generates an approximate iterative solution $\mathbf{u}_{(i+1,j)}$ converging toward $\mathbf{u}_{(i+1)}$.

The choice of preconditioner is clearly important. In fact, some of the previously discussed algorithms require efficient preconditioners for both sides of the problem; i.e. for both the plasma and the neutral equations. An example of a potentially useful preconditioner for the plasma side is to solve the plasma equations using a fluid neutrals model. Of course, even simpler models (enhanced plasma diffusion?) are preferable if they are found to work efficiently. This implies that the solution strategy for Eq. 34 is to: (i) solve the fluid neutrals-fluid plasma problem as an initial guess for each time step and (2) use the fluid neutrals-fluid plasma approximation for the Jacobian where needed in the various algorithms to follow.

B. Transparent

If the neutral kinetic equation is solved in a manner that is conducive to transparent coupling, then the best JFNK approach is likely to be solving the full plasma-neutral problem. Although the system has high dimensionality, the equations have a known sparsity pattern that can potentially be exploited. To the extent that the individual plasma and neutrals solver already contain implicit nonlinear solvers, good preconditioners may already be known for each part of the problem.

The neutral kinetic equation for $\mathbf{u}_D = f$ is of the form

$$\mathcal{N}_D + \mathcal{N}_C = \partial_t f + \nabla \cdot \vec{v} f + \nabla_v \cdot \vec{F} f - C[f, \mathbf{u}_A] - S[f, \mathbf{u}_A] \tag{37}$$

where C is the collision operator, S is the ionization/recombination source and F represent any forces on neutrals. The evaluation of the kinetic equation must be defined over time span Δt. The JFNK solver determines iteration $f_{(i+1,j)}$ given value $f_{(i)}$. The MC code must be used to find $\partial_t f_{(i+1,j)}$. For long times, the value of the kinetic equation is best defined via

$$\mathcal{N}_C + \mathcal{N}_D = \left(f_{(i+1,j)} - f_{(i)} - \int_0^{\Delta t} dt \partial_t f_{(i+1,j)} \right) / \Delta t. \tag{38}$$

The partial derivates needed for the Jacobian of the kinetic equation can be almost completely determined analytically. Only the evaluation of non-constant terms in the Jacobian, which arise from nonlinear terms in the kinetic equation, must be performed by insertion of the numerical solution for $f_{(i+1,j-1)}$.

The electrostatic quasi-neutral plasma fluid equations $\mathcal{N}_A + \mathcal{N}_B$ for variables $\mathbf{u}_A = \{N_i, V_i, T_i, V_e, T_e, \phi, \varpi\}$ are given by the fluid moment equations. For example, for ions

$$\partial_t N_i + \nabla \cdot N_i V_i = S_{i,0}(\mathbf{u}_A, f) \tag{39}$$

$$\partial_t N_i V_i + \nabla \cdot (N_i V_i V_i + P_i) = R_{i,1}(\mathbf{u}_A, f) + S_{i,1}(\mathbf{u}_A, f) \tag{40}$$

$$\partial_t \tfrac{3}{2} P_i + \nabla \cdot (\tfrac{3}{2} P_i V_i + Q_i) + P_i \nabla \cdot V_i = R_{i,2}(\mathbf{u}_A, f) + S_{i,2}(\mathbf{u}_A, f). \tag{41}$$

where R represents collisional friction and S represents ionization and recombination sources. A similar equation set holds for electrons, along with the constraint $N_e = \sum_i Z_i N_i$ and the "vorticity" equation for the potential which is defined via the current continuity equation $\nabla \cdot \mathbf{J} = 0$. In these equations, the velocity perpendicular to the magnetic field $\mathbf{B}$ is defined to lowest order by the drift motion for each species defined by the balance between the Lorentz force $0 = \mathbf{F}_i + Z_i e \mathbf{V}_i \times \mathbf{B}$; the drift flow is $\mathbf{V}_{i,\perp} = \mathbf{F}_i \times \mathbf{b} / Z_i e B$ and, for the electric field one defines $\mathbf{V}_E = \mathbf{E} \times \mathbf{b} / B$. The sum of the force balance equations allows one to solve for the perpendicular current $\mathbf{J}_\perp = \sum_i \mathbf{b} \times \mathbf{F}_i N_i / B + \ldots$. The divergence of the current is formally one order higher, requiring that inertia be retained for the charge conservation equation. The continuity equation is therefore

$$\partial_t \varpi + \nabla \cdot \mathbf{V}_E \varpi = \nabla \cdot \sum_i \mathbf{F}_i \times \mathbf{b} / B + \nabla \cdot J_\parallel \mathbf{b} + \nabla_\perp^2 \mu \varpi \tag{42}$$

where

$$\varpi = \nabla_\perp \cdot \left[\sum_i Z_i e N_i \rho_i^2 \left(T_i^{-1} \nabla_\perp Z_i e \phi + p_i^{-1} \nabla_\perp p_i \right) \right] \tag{43}$$

$$\rho_i^2 = M_i T_i / (Z_i e B)^2. \tag{44}$$

C. Semi-Transparent: Constrained

If the plasma data is accessible (system $\mathcal{A}$), but the neutrals are too difficult to access (system $\mathcal{D}$), then the problem is semi-transparent. The kinetic neutral equation can be handled as a constraint, which means solving the kinetic equations

$$\mathcal{N}_D(\bar{f}_{(i,j)}) + \mathcal{N}_C(\bar{f}_{(i,j)}, \mathbf{u}_{A,(i,j)}) = 0 \tag{45}$$

for every JFNK subiteration (with index j) for each time step (with subindex i) that is required. It is clear that, for large time steps, this is expensive and that it will induce a dense Jacobian factor into the plasma equations. The largest time-step that can be taken will be determined by the ability to find a good preconditioner for the plasma system

$$\mathcal{N}_A(\mathbf{u}_{A,(i+1,j)}) + \mathcal{N}_B(f_{(i,j)}, \mathbf{u}_{A,(i,j)}) = 0. \tag{46}$$

D. Semi-Transparent: Reduced

In order to generate a system with a simpler sparsity pattern, let us define the subsystem of fluid neutral variables $\mathbf{u}_E = \{N_n, V_n, T_n \dots\}$ and moment equations (as in Eq. 39-41) which have the structure

$$\mathcal{N}_{D'}(\mathbf{u}_E) + \mathcal{N}_{C'}(\mathbf{u}_E, \mathbf{u}_A) = \partial_t \mathbf{u}_E + \mathcal{F}_E(\mathbf{u}_E) + \mathcal{N}_{C'}(\mathbf{u}_E, \mathbf{u}_A). \tag{47}$$

The JFNK solver will now have access to plasma and neutral fluid equations.

How shall the neutral distribution function be defined for a given set of moments? We require two novel uses of the kinetic equation: (i) we must evaluate the kinetic equation *for a specified set of fluid moments* and (ii) we must compute the Jacobian with respect to the moments for a given f. To partially constrain the system, we define the local Maxwellian

$$f_M = \exp\left(-m(v-V)^2/2T^2\right)/(2\pi T)^{3/2} \tag{48}$$

and the moments defined by the nonlinear constraints

$$N = \int f d^3v \tag{49}$$

$$NV = \int v f d^3v \tag{50}$$

$$NT = \int v^2 f d^3v/3. \tag{51}$$

and so on.

The collection of moments is given by the n_E coordinates $\mathbf{u}_E$ and the auxilliary constraints are defined by the solution for the variables $\mathbf{u}_F \equiv \delta f = f - f_0$, where f_0 has been determined by the moment equations. Standard choices of orthogonal polynomial representations can be used for f_0, where

$$f_0 = f_M \sum_{k=0}^{n_E-1} a_k(x,t) P_k((v-V)/v_T) \tag{52}$$

$v_T = \sqrt{T/m}$, the $P_k(x)$ are a set of orthonormal polynomials ($a_0 = N$), and the coefficients are related to moments of the distribution. The distribution f_0 acts as a source that depends solely on the fluid moments ($\mathbf{u}_E$) and acts as a driver for the nonlinear constraint equation

$$\mathcal{N}_F[\delta f, \mathbf{u}_A] = -df_0/dt. \tag{53}$$

The solution to this equation must satisfy the auxilliary constraints $\delta N = \delta NU = \delta NT = 0$. This is precisely true when the moments satisfy the moment equations leaving only the remainder

$$df_0/dt = \partial_t f_0 + \nabla \cdot v\delta f_0 + \nabla_v \cdot F f_0 \tag{54}$$

$$= \sum_{k=0}^{n_E} \left[f_M a_k \int P_k(C+S) d^3v + f_M v \cdot \nabla a_k P_k + a_k F \cdot \nabla_v P_k f_M \right]. \tag{55}$$

Thus, the sources are defined so as to generate no source correponding to each orthogonal basis function in the moment expansion. In practice, an MC approach may need specialized techniques to eliminate changes to the moments induced by noise.

The Jacobian can be performed by a combination of analytic and numerical differentiation. The result is the linearized moment equations. For example, particle conservation is expressed as

$$\partial_t \delta N + \partial_x (V\delta N + N\delta V) = \sum_{G=A,E} \delta \mathbf{u}_G \left[\int d^3v (\partial S/\partial \mathbf{u}_G) + \int d^3v (\partial S/\partial \delta f)(\partial \delta f/\partial \mathbf{u}_G) \right]. \tag{56}$$

The other moment equations have a similar form, where, now collisional terms are also introduced. Only the source terms need to be computed numerically. In principle, the partial derivative of the first source term, or at least the majority of its dependence on the fluid moments, can also be computed analytically. Since these term may depend on δf, it will ultimately require the numerical result for evaluation, but not for differentiation.

E. Opaque

If the system is treated as if it were completely opaque, then the approach to iteration will rely on the ability to solve the neutral kinetic equation efficiently for long time steps. As discussed in the previous sections, a neutral fluid model can be used as the preconditioner need to invert the plasma coupling equations: $\hat{\mathbf{A}}_0 = \mathbf{A} - \mathbf{B}\mathbf{D}'^{-1}\mathbf{C}'$. It is also prudent to identify an inexpensive model for plasma response $\mathbf{A}_0$ for use in the neutral kinetic equation inversion: $\hat{\mathbf{D}}_0 = \mathbf{D} - \mathbf{C}\mathbf{A}_0^{-1}\mathbf{B}$. This term generates anisotropy in the neutral kinetic equation due to neutral-plasma interactions. The number of iterations can be reduced if the preconditioner is a reasonable predictor of the long-time plasma response.

The plasma fluid equations are distinguished by high anisotropy with respect to to the magnetic field. For example, parallel conductivities typically smooth the parallel dependence of temperature and velocity along the field lines. For example, the response of the plasma temperature to a source is approximately determined by the Braginskii thermal conductivity

$$B_0\partial_\ell(\kappa_0/B_0)\partial_\ell T_A^{5/2}\delta T_A = (\partial Q/\partial \mathbf{u}_D)\delta\mathbf{u}_D + \dots. \tag{57}$$

The main response is then

$$X = \int \{(\partial Q/\partial \mathbf{u}_D)\delta\mathbf{u}_D\}\, d\ell/B_0 \tag{58}$$

$$\delta T_A = T_A^{-5/2}\int \left\{X B_0^2/\kappa_0\right\} d\ell/B_0 \tag{59}$$

where $\{X\} = X - \langle X\rangle$ and $\langle X\rangle = \oint X d\ell/B_0$ is the average along a field line. A similar result holds for the parallel plasma velocity. The more challenging aspect to model is the plasma density response. It is often the case that the parallel velocity dominates in the neutral-plasma interaction region, because the plasma is accelerated to a fraction of the sound speed near the ionization front. In this case, one can solve

$$\partial_\ell\delta(N_A V_{A\|})/B_0 = S_0 \tag{60}$$

$$\delta(N_A V_{A\|}) = B_0\int \{(\partial_D S)\delta\mathbf{u}_D\}\, d\ell/B_0. \tag{61}$$

However, the parallel velocity can vanish deep into the neutral region due to collisional friction.

IV. CONCLUSION

This article has emphasized the utility of the JFNK approach for solving the code coupling problem. Various levels of communication capability require different choices of algorithms. The less communication that is available, the more dense the Jacobian matrices become and new types of preconditioners must be sought to efficiently take large time steps. The more communication that is available, the more possible it becomes to exploit the simple sparsity pattern of the Jacobian, albeit of a larger system.

ACKNOWLEDGMENTS

The author would like to thank J. W. Banks, A. M. Dimits, L. L. Lodestro and T. D. Rognlien for useful discussions. Work for this technical report LLNL-TR-654916 was performed under the auspices of Lawrence Livermore National Laboratory under U.S. D.O.E. contract DE-AC52-07NA27344.

[1] D. A. Knoll and D. E. Keyes, J. Comp. Phys. **193**, 357 (2004).
[2] D. A. Knoll, V. A. Mousseau, L. Chacón and J. Reisner, J. Sci. Comp. **25**, 213 (2005).
[3] D. P. Stotler, C. F. F. Karney, M. E. Rensink, and T. D. Rognlien, Contrib. Plasma Phys. **40**, 221 (2000).
[4] S. I. Braginskii, *Transport Properties in a Plasma, in Review of Plasma Physics*, M.A. Leontovish (ed.), Consultants Bureau, vol. 1, pp. 205-311 (1965).
[5] T. D. Rognlien, J. L. Milovich, M. E. Rensink, and G. D. Porter, J. Nucl. Mater. **196-198**, 347 (1992)

[6] T. D. Rognlien, D. D. Ryutov, N. Mattor, and G. D. Porter, Phys. Plasmas **6**, 1851 (1999).
[7] D. Heifetz, D. Post, M. Petravic, J. Weisheit, and G. Bateman, J. Comp. Phys. **46**, 309 (1982).
[8] D. P. Stotler and C. F. F. Karney, Contrib. Plasma Phys. **34**, 392 (1994).